フレッシャーズ特別企画

第2章

1+1を10にする「チーム力」養成講座

意見を戦わせより良い成果を生み出そう

穴田啓樹

ここでは，チームの力を発揮するためのコミュニケーションについて考えます．一般的なコミュニケーション理論の説明ではなく，筆者が新しいチームを立ち上げてから2年にわたって実践してきたことを，その中でも特に効果のあったものを中心に紹介します．（筆者）

1 チーム開発に加わる

ここでは入社1年目～3年目程度の皆さんを対象に，チームで開発を進める際の基本的な考え方を説明します．

● より良い成果を生み出すのが「チーム開発」

例えば，学生時代に取り組んだ卒業研究を思い出してください．多くの場合，担当教官から一人ずつ異なるテーマを与えられ，一人で研究を進め，一人で論文にまとめたことと思います．つまり，卒業研究では一人一人の研究成果が評価されるわけです．

これに対して会社では，大きなテーマを多人数で担当し，一人では成し得ない大きな成果を生み出します．プロジェクト・チームを立ち上げ，役割を分担して開発を進め，それを全体としてまとめ上げていきます．

ただし，人が集まるだけですばらしい成果が出せるかというと，そうではありません．ここでは「チーム開発」を次のように考えます（**図1**）．

- グループ開発：人が集まり分担して作業している状態
- チーム開発：メンバが互いに良い影響を与え合い，全体としてより良い成果を生み出している状態

● 今こそ「チーム力」が必要とされている

筆者は近年，チームの力を発揮する必要性を今まで以上に強く感じています．それはなぜでしょうか．私たちを取り巻く環境の変化を振り返ってみましょう．

1）開発規模が大きく複雑になってきている

開発規模が大きくなり，より複雑になった結果，私たち

図1　お互いに良い影響を与え合うのが「チーム開発」

KeyWord　チーム開発，コミュニケーション，伝え手の責任，伝える目的，場を感じる，コーチング，意識付け，意味付け，テクニカル・ライティング，即題スピーチ，ペア開発，4行日記

が直面する課題はより難しくなっています．その課題に取り組むには，各部分の担当範囲だけに目を向けるのではなく，システム全体を見据えた上で開発者が協調することが求められています．

2）技術が高度に専門化している

技術が高度に専門化した結果，開発リーダがすべての詳細を把握して指示を出すというスタイルが成立しなくなっています．一人一人のエンジニアが担当部分に関して高い専門性を持って主体的に取り組むこと，また，それを全体として有機的にまとめ上げることが求められています．

3）より優れた発想が求められている

会社に入ると，なぜ給料（お金）をもらえるのでしょうか．その仕組みを突き詰めていくと，自分たちの活動が社会に価値を生み出し，お金に替わることが分かります．その知的活動の原点は発想にあり，個人が独自の発想を持つことが大切です．

しかし，個人の発想や思考には限界もあります．そこで，チーム・メンバが互いの考えをぶつけ合うことで，より良い発想に育てていき，社会に競争力のある商品やサービスを提供していくことが求められています．

2 チームを伸ばすコミュニケーション

チームの力を発揮するためには，チーム・メンバ間のコミュニケーションを促すことが重要です．では，いったいどのようなコミュニケーションが必要なのでしょうか．開発プロセスやプロジェクト・マネジメントの解説書でもコミュニケーションの重要性が説かれていますが，具体的にどうすればよいのかは，書かれていないことが多いようです．ここでは，コミュニケーションを円滑にするための，七つの基本的な考え方を紹介します．

● その1：まず，自分の考えを伝えること

チームのコミュニケーションは，まず，自分の考えを相手に伝えることから始まります．自分から相手へ情報を発信し，その情報に相手が反応します．連鎖反応を引き起こしながら，情報の価値を高めていくのです．このサイクルをきちんと回しながら開発を進めることが重要です．

仕事では，「そんなこと，わざわざ言わなくても分かってくれるだろう」などと，相手の「察し」を期待してはいけません．多くの場合，自分の期待と相手の考えは異なります．そのギャップがトラブルの原因になるのです．

結果が良くないときには，伝えにくいこともあるでしょう．そのようなときでも，きちんと伝える習慣を付けましょう．

● その2：伝え手が責任をとること

「ドキュメントを読んだが内容を理解できなかった」．そんな経験を持つ読者は多いでしょう．コミュニケーションの責任，つまり「情報が伝わる」ことは誰が受け持つべき責任なのでしょうか．

一昔前は，受け手の責任とされてきました．「君はそんなことも分からないのかね」，「それは常識だよ」といった会話が日常的に交わされてきました．しかし，この十数年で，それは伝え手の責任に変わってきています．開発の大規模化と社会の雇用形態の変化によって，多様なメンバが開発に加わるようになり，これまで「常識」とされてきたことが通用しなくなっています．その結果，伝え手に「相手が理解する」ことを意識したコミュニケーションが求められているのです．「伝えたか」ではなく「伝わったか」が重要なのです（**図2**）．

これに気付いていない組織では，情報の受け手の時間を浪費しています．例えば，本来，10分で内容が伝わる報告書があったとします．受け取った側がそれを読むのに30分かかったとすると，20分の時間を無駄に使ったことになります．読み手が10人ならば200分の無駄になります．逆に5分で内容を理解できれば，50分の時間を節約したことになります．これが日々積み重ねられるとしたら，大変な格差を生みます．私たちが発信する情報は，チーム・メンバに影響を与えていることを自覚しましょう．

図2 「伝わらない」のは誰のせい…？

● その3：伝達する目的を明確にすること

コミュニケーションには目的があります(図3)．なぜ相手に情報を伝えるのか．この「なぜ」の深さが浅い段階で情報を発信すると，受け手は困惑します．

例えば，あなたが開発リーダに「xxxの新しいアルゴリズムを考えました」と技術的な詳細を延々と報告したとしましょう．リーダは，あなたが新しい案を考え出したことは評価するかもしれません．しかし，その案に関心がなければ話はそこで終わってしまいます．それどころか，忙しくて時間のないときは最後まで聞いてもらえないでしょう．

あなたは，なぜそれを伝えたかったのでしょうか．もし，あなたが「新しいアルゴリズムを評価する時間が欲しい」のであれば，「承認してほしい」というコミュニケーションの目的を伝えなければならないのです．

この例の場合なら，報告の冒頭で「新しいアルゴリズムの評価を承認していただきたいので」と，話の目的を伝えるのです．最初に目的を伝えるだけで，その後に続く技術的な内容をスムーズに伝えることが可能になります．

● その4：事実と意見を区別すること

次の三つの文章の違いを考えてください．

1) それは…です．
2) それは…だと思います．
3) それは…だと考えられます．

普段，これらの言葉の違いを意識せずに使っているとしたら，あなたはチーム・メンバを困惑させたり，誤った情報に振り回されたりしているかもしれません．なぜならば，情報の正確さが異なるからです．**1)** はより事実である可能性が高く，**2)** と **3)** は意見にすぎません．

開発上の課題を解決する道筋を考えるにあたり，起点にする情報を誤ると，ゴールに到達できません．ですから，情報を受け取ったら，まず，それは事実なのか意見なのか，さらに，誰が言っていることなのか，根拠はあるのかなどを確認しましょう．

● その5：理解を確認すること

人が外界から情報を得てから理解するまでの過程は，図4のモデルで考えることができます．チーム開発を進める上で気を付けなければならないのは，すべてのチーム・メンバが同じ復号器を持っているわけではない，ということです．打ち合わせで同じ話を聞いたり，同じドキュメントを見たりしても，理解の程度がおのおののチーム・メンバで異なる可能性があるのです．

ですから，自分が理解したと思っていることを，必ず相手に確認しましょう．何か説明を受けたときには，自分なりに解釈して，それを自分の言葉で相手に説明してみましょう．その説明についてチーム・メンバに聞いてもらい，理解が正しいかを確認します．このサイクルを繰り返すことによって，お互いの理解が近づいていくのです．

● その6：対立を恐れず，より良い発想を求めること

会議で「間違っていたら恥ずかしい」，「こんなことを言うと雰囲気が悪くなるんじゃないか」と感じたことのある読者はいませんか．そのようなときは，何に意識が向いているでしょうか．

人に意識が向かうと感情が生まれ，伝えにくくなります．コミュニケーションの目的を思い出しましょう．会議の目

図3 「伝える目的」を伝えることが重要

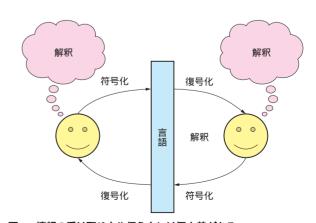



図4 情報の受け取り方や伝え方には個人差がある
情報を伝える際や受け取る際には，その人固有の解釈が入る．

的は，より良い発想を生み出すこと，課題，問題を解決することです．そのために，意見に意識を向けて，意見同士を戦わせるのです．「自分はより良い結果を生み出したいのだ」．会議ではいつもこのことを思い出しましょう（下掲のコラム「意見の衝突が成果につながった合同新人研修」を参照）．

なお，そのように安心して意見を戦わせる雰囲気（場）を作るのが，リーダの役割になります．

● その7：柔軟でいること

開発チームの中には多様な考え方を持つメンバがいるでしょう．あなたは，もし自分と異なる意見に触れたとき，どのように反応するでしょうか．

もし，その意見を拒否してしまっていたとしたら，それはもったいないことをしています．新しい考えに触れたときには，その善しあしを判断する前に，まずは受けとめてみましょう．人は善しあしを判断すると，その判断に従って考える傾向があるので，情報の解釈を誤る可能性があるからです．

どんな意見でも，「もしかしたら参考になるところがあるのではないか」と素直に考えられると，より豊かな発想ができるようになります．多様な意見を受けとめるようにしましょう．

3 チーム開発をリードする

ここでは入社5年目以降のエンジニアの皆さんを対象に，チーム開発を促すことについて考えます．入社して5年もたつと開発チームのリーダを任され，立場の変化に戸惑う方もいるでしょう．ここで5分ほど時間をとり，これまでの開発の場面を思い出しながら振り返ってみてください．

あなたが自分の仕事に専念できていたとき，周囲の人はあなたにどのようにかかわっていたでしょうか．逆に，仕事に集中できなかったとき，何があなたを妨げていたのでしょうか．ここでは，チームの力を発揮するために筆者が最も心がけている七つの項目を紹介します．

● その1：場を感じること，場を作ること

リーダの最も重要な役割は，場を感じ，場を作ることです．まず，「場を感じる」ことから始めます．一人一人のメンバが発する言葉の意味，声の調子，表情，姿勢から，職場を見て漠然と感じることまで，あらゆることやあらゆる変化を敏感に感じ，洞察しなければなりません．そこから感じることは開発の行方に何らかの影響を与え，何らかの意味を持っているからです．

「場を感じる」というと難しいことのように思えますが，プロジェクトのことを深く考え続けること，チーム・メンバに対して強い関心を持ち，集中して会話していくことで，

COLUMN

意見の衝突が成果につながった合同新人研修

組み込みコミュニティのご縁で，2005年5月から1カ月にわたり，富士通デバイスと筆者の会社（キャッツ）で合同の新人研修を試行しました．対象となった新入社員は，合同研修の開始時点で，既に各社の1カ月間の研修を受けていました．入社後1カ月しかたっていなかったのですが，既に各社それぞれの雰囲気ができていました．そして，その両社のメンバが入り交じってチームを組み，組み込みシステム開発の一連の流れを体験していきました．

中盤には率直に議論を交わし合うようになり，意見が衝突することもありました．しかし，その衝突からより良い結果が生み出されることを目の当たりにして，衝突を恐れない姿勢の重要性を再認識しました．以下，参加者を代表して，6名の方の研修の感想を掲載します．

- お互いに意見を出し合い，検討，協力して最善だと思われる構成を導き出し，解決したことが貴重な経験となった．（梶本貴幸氏）
- さまざまなものの考え方や，それらをまとめてゴールへ向かう合同開発の楽しさと難しさを学ぶことができた．（渡邉幸嗣氏）
- 業種の異なる企業の方と合同で研修を行うことにより，業種間のスタイルの違いなどを感じることができて，とても刺激になった．社内研修とは違い，良い意味でライバルみたいな関係で高め合えたのではないかと思う．（川中亮輔氏）
- 異なる会社の人ということで，同じ会社の人とは視点が違った考えがいくつか出てきて，大変勉強になった．（太田和憲氏）
- チーム開発の難しさと，チームで問題を解決できたときの喜びを経験できた．二つの会社の個性や得意分野を生かして組み込みシステム開発の流れを体験できたのが貴重な体験となった．（田村政和氏）
- メンバの意見を真に理解することの難しさ，自分の意見を伝えることの難しさを感じた．また，仲間とのコミュニケーションの重要性を学んだ．（松永好郎氏）

徐々に感じる能力が高まってくるのです．
次に，「場を作る」ことですが，これはメンバからの信頼の上に成り立ち，リーダとしてのメンバへの影響力から生まれるものです．信頼を築くのは時間がかかるものですが，チーム・メンバに関心を持つことや個々のメンバが一緒に働く意味を考えることから始めましょう．個々のメンバは現在，チームに対してどのような影響を与えているでしょうか．これから共通の目標を達成するために，どのような力を発揮し得るでしょうか．また，どのような状況で力を発揮しやすいのでしょうか．
また，チーム・メンバ同士が互いの価値観を共有できる機会を設けましょう．例えば，不具合を解決していく過程で，個々のメンバが普段は何に重点を置いて原因を調査しているか，不具合を作り込まないように何に気を付けているかなど，一人一人に意見を求めます．お互いに大切にしていることや判断基準を知ることによって，チーム・メンバを尊重する雰囲気を作り上げるのです．
そのようにチーム・メンバが安心して力を発揮できる場を提供できるリーダには，自然と信頼感がついてきます．

● その2：チームのシナリオを描く

開発がゴールにたどり着くまでのシナリオ（筋書き）を描き，メンバに語りかけましょう．人はシナリオ，すなわち文脈から理解しやすいものです．シナリオは，開発の初期に一度伝えるだけでは不十分です．毎日，毎週，毎月．その時その時のマイルストーンを達成し，問題を解決していくシナリオを共有していきましょう．
このシナリオ共有が不十分だと，メンバはどこに向かっているのか分からなくなり，不安を感じます．その結果，チームの力を発揮しにくくなるのです．
シナリオは，まず目指す状態，あるべき姿から始めます．そして，現状とのギャップを明確にして，中間地点を設けながら，いつ，誰が，どのように課題に取り組み，その結果どうなるのかを説明します．そして，目標を達成することの意味，チーム・メンバへの思いを語りかけましょう．この語りかけがシナリオのイメージをより具体的にするのです．また，メンバの反応を見ながら，相手に届きやすいメッセージを選ぶように心がけましょう．

● その3：自ら話を聞いて回る

チーム・メンバの報告を待つのではなく，リーダから進んで状況を把握するためにインタビューに回りましょう．自分から話を聞いて回ることによって，チームの状況を直接「感じる」のです．状況を把握するときは「事実と意見を区別すること」で説明した通り，メンバとの会話から事実を引き出すことが肝心です．このとき，リーダから見ておかしいと考えられる言動があっても，頭ごなしに否定せず，まずは意図を確認してみましょう．意図は同じなのに言葉のあやに引っかかることはよくありますが，お互いの意図が同じであることが確認できれば，スムーズに話を進められます．
また，単に進ちょくを確認する会話では尋問になり，ギクシャクした雰囲気になってしまいがちです．そこで，その仕事の難しかった点，工夫した点，自己評価を聞いてみるとよいです．そして，「そういう点が難しいんだね」，「面白い工夫をしたんだね」といったような，相手の行為を認める会話を進めましょう．その言葉からメンバは，「リーダは自分のことを見てくれている」という安心感を得ることができ，よりチームの力を引き出すことができるのです（左掲のコラム「コミュニケーション・スキルの磨き方」を参照）．

● その4：メンバを巻き込む

開発上の課題や問題はメンバの力を引き出し，巻き込みながら解決していきましょう．リーダが「何とかしてしまう」のではなく，チーム・メンバが主体となり，解決していく機会を増やすのです．「自分たちが開発を支えている」

COLUMN
コミュニケーション・スキルの磨き方

近年，ソフトウェア開発でもコーチングが注目されるようになりました．職業コーチになるわけではないにせよ，その考え方やコーチング・マインドを持ってコミュニケーションすることは，チーム開発にプラスに作用します．セミナを受講したり書籍を読んでみたりして，いざ実行しようとすると，初めのうちは恥ずかしさを感じるものです．例えば，今までは仕事の進ちょくを「終わったか」，「終わっていないのか」しか確認していなかった状況で，「これは随分と複雑なアルゴリズムだったから，実装を終えることができてうれしそうだね（笑顔）」などと，反映的傾聴するのは勇気がいるでしょう．
しかし，自転車に乗れるようになったときと同じで，最初からうまくできるものではありません．実践していく中で磨かれるものですから，あせらずにスキルアップを目指しましょう．

という意識は，チームの力をより強固なものにします（下掲のコラム「ホワイト・ボードの活用」を参照）．

● その5：メンバの不安を取り除く

人は不安があると行動しにくくなるものです．そこで，開発リーダは普段からチーム・メンバが不安を抱えていないかどうか，話を聞いて回ります．そして可能な限り，その不安を取り除くように支援します．

まずは事実の確認から始め，どこに不安を感じているのか明確にしていきます．漠然として言語化できないからこそ不安に感じるものですから，会話を進めながら，一緒に探り当てていきましょう．抱えている不安を聞いてもらえるだけでも随分と楽になるものですが，それを取り除いていく際には，視野を広げて不安を感じている部分を俯瞰（ふかん）したり，見方を変えたりする会話が有効です．

また，人は自分の能力には気付きにくいものなので，リーダがメンバの能力を思い出させてあげることが有効です．「できるかもしれない」と思うと勇気がわいて，次の行動に移れるようになります．その上で，具体的な話を進めると効果的です．

● その6：仕事の意識付け/意味付けを行う

意識付け，意味付けは，チーム・メンバの成長を促進するために行います．仕事を始める前に「意識付け」，仕事を完了した時点で「意味付け」を行います（**図5**）．「意識付け」は，その開発を通して身に付けられる技術と，身に付ける意味について，その技術を身に付けたメンバ自身をイメージできるように会話を進めます．この意識付けを行うことにより，身に付けたい技術が開発を通してより染みこむようになります．「意味付け」は開発の振り返りと同時に行い，開発を通して体験した技術をスキルとして定着させるために行います．

まず，開発を通してどんな体験をしたのか思い出すように会話を進めます．そして，そこから何を学んだのか，それは今後のチーム・メンバにどのような影響を与えるのかを話し合います．これらを繰り返すうちに，「なんとなくできていたこと」が「きちんと意識してできること」になり，自信を持てるスキルになります．

● その7：リーダ自ら成長する

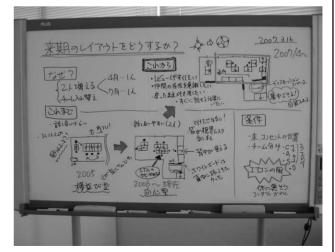
より大きな目標に取り組めるようにチームを成長させるのは，リーダの役割です．では，リーダはメンバの成長を見守るだけでよいのかというと，決してそうではありません．むしろ，リーダ自身の成長意欲がチーム・メンバに波及し，メンバの成長を促していくのです．

成長意欲のないリーダのチームは，どのような状況に陥るのでしょうか．例えば，ある専門性の高い仕事があり，ある時点でリーダしかできなかったとします．そして，月

COLUMN
ホワイト・ボードの活用

打ち合わせして，すべて明確にしたつもりが，いざ行動に移そうとした瞬間に分からないことだらけになることがあります．このようなことをなるべく防ぐために，筆者のチームでは，ホワイト・ボードを使って小まめに打ち合わせしています．

一人ずつペンを持って，会話しながら書き進めていきます．相手がいないときには，一人でホワイト・ボードを通して自分と対話します．すると不思議なことに，書き出してみて気付くことが結構あるのです．このことに気付いてから，筆者のチームでは，とにかく書き出してみてから考える習慣にしています．

進め方のポイントとしては，話し合いの経過が分かるようにします．結果だけが残っていても，なぜそうなったのか経過が分からないと，議論をぶり返すことが多いからです．また，普通は書き出さないような「頭の中のつぶやき」や，そのときの「思いの言葉」を残すようにしてください（**写真A**）．それらが発想を広げてくれたり，手がかりになっていきます．

写真A　ホワイト・ボードの活用例
来期のレイアウトについて，5人で40分ほど議論しながらまとめたもの．

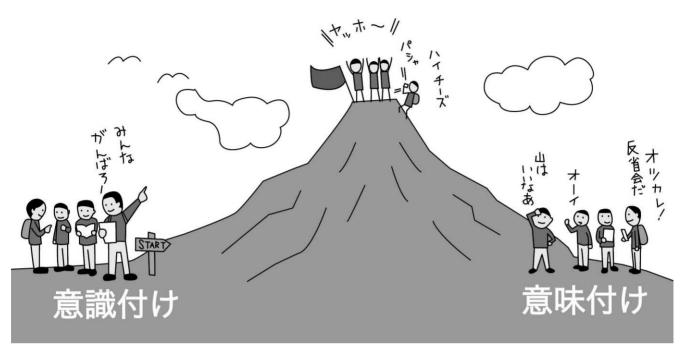



図5　仕事の前に「意識付け」を，仕事の後に「意味付け」を

日がたち，後輩が育って，その仕事を担当するに十分な力を付けるようになったとしましょう．もし，そのような状況になってもなお，リーダが「これは自分しかできない仕事だ」とそこにとどまり，手放さないのであれば，後輩は新しい仕事に挑戦する機会を失います．その結果，玉突き衝突を起こしてチーム全体の成長が止まってしまうのです．

よって，リーダは自ら新しい領域の仕事に挑戦し，リーダの仕事をメンバに任せることにより，チーム全体が伸びていくように環境作りを心がけましょう．

4 あるチームの風景

ここでは筆者の職場で実際に行っている，チーム力を向上するための実践例を紹介します（**図6**）．これらは，チームの力を発揮する上で筆者が必要だと感じて経験的に始めたことであり，内容や進め方はチームの進化に合わせて常に変化させています．皆さんの参考になるものがあれば，アレンジして利用していただければ幸いです．

● 1日リーダ

その日の開発，特に定常業務がスムーズに行えるように，リードする人を決めます．小学校のころの「日直」のようなものだと考えてください．リーダだけがリードするのではなく，メンバの一人一人がリーダシップを発揮する場を作ります．

● 朝礼

1日リーダが司会を務めながら，メンバの中に，その日を終えたときの仕事や人の状態のイメージを作っていきます．これは，「何をしました」とか「何をします」といったことを報告する場ではありません注1．あくまで1日を終えた時点での成果の状態，誰が何をどのレベルで達成するのかを明確にします．このとき1日リーダは，目標を達成したメンバ自身をイメージするような，リーダ自身の言葉をかけるようにします．「これまで緻密に分析を進めてきたから，結果がOKになればうれしいよね」といった具合です．この達成イメージを持つだけで，その日1日の時間の使い方に随分と差が出てきます．

また，1日リーダが持ち回りで司会を務めることで，自分の仕事だけでなく，ほかのメンバの仕事にも視線を向けることができます．仕事の全体像と自分の仕事との結びつきを感じることによって，チーム感が醸成されます．

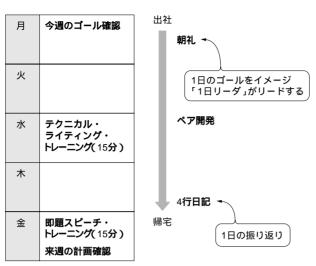



図6　筆者のチームで実践している例
1日の中でやることと，1週間の中でやることをそれぞれ設けている．

図7　テクニカル・ライティング・トレーニング
1人が参考書の一節を朗読し，要約する．ほかのメンバが思い当たることを話し，メンバ間で共有する．それぞれの役割は持ち回りで担当する．

● テクニカル・ライティング・トレーニング

前述の「伝え手が責任をとること」で述べたように，ドキュメントやメールの書き方次第で相手の時間を奪ってしまう可能性があります．そこで，文章の書き方とその効果について考えるトレーニングを行っています（**図7**）．

参考文献（1）をテキストとして進めています．具体的には，まず，その日の朗読担当が文献の一節（1ページ程度）を音読します（朗読）．次に，自分が理解したことを自分の言葉でメンバに説明します（要約）．そして事例担当が，その日の朗読分に関して，これまでの自分の体験から思い当たることを話し，メンバ間で共有します（事例）．最後に，事前に予習しておいた解説担当が，今後の行動に結びつけるようにコメントします．これが通常の流れですが，時には，実際に職場で流れた誤解を招きやすいメール，理解しにくい仕様書，逆に理解しやすい仕様書を題材に，なぜ分かりにくいのか，分かりやすいのかを考える場にしています．

このトレーニングは，週に1度，1回あたり15分，と短時間で進めています．1年も経過すると，参加者たちの書くメールや仕様書に随分と効果が表れてきました．また，分かりやすい書き方について，お互いに意見を出し合えるようになりました．

● 即題スピーチ・トレーニング

テクニカル・ライティングは「書く力」を伸ばすものでしたが，「即題スピーチ」は話す力を伸ばすために行っています．時間が限られている開発の中では，要点をまとめて話すことが求められます．そこで，時間を1分，3分と決め，要約して話すトレーニングを行うのです．いつも同じ時間ではなく，今日は1分，今日は3分，などと変えることによって，話の構成を意識するようになり，まとめる力がさらに伸びます．また，ストップウォッチで実際の時間を計測することで，「時間が経過する」という感覚が身に付きます．時間が限られていることを認識し，時間の経過を感じることができるようになると，ミーティングの質が随分と向上しました．

さらに，初めは事前に決めたテーマで話すのですが，慣れてきた時点で，テーマは直前に決めるようにします．どんなテーマになるか分からないので，その場に対応する力が磨かれます．

この「即題」方式を3カ月ほど続けた時点で，別の効果に気付きました．その場でテーマを決めることにより，その人の考え方のよりどころや大切にしていることが現れやすくなるのです．それを共有する場ができた結果，お互いのコミュニケーションがスムーズになりました（p.112のコラム「自己紹介ワークショップ」を参照）．

● ペア開発

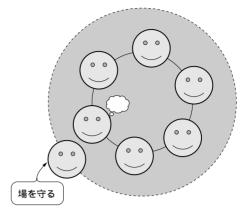
「ペア開発」とは，アジャイル・プログラミングのプラクティスとして知られている「ペア・プログラミング」をプログラミング以外の工程でも活用するものです．これは笑い

注1：ただし，週始めなどは，先週の「頭」を取り戻すために，状況を1分で要約してもらうこともある．

COLUMN
自己紹介ワークショップ

チーム開発の原動力はチーム・メンバの多様性，多様な発想にあります．それを開発力につなげるには，まず，チーム・メンバにそのことを気付いてもらうことから始めます．そのきっかけの作り方の一つとして「自己紹介ワークショップ」(**図A**)を紹介します．例えば，新しいチームを立ち上げるにあたりチーム・メンバが初めて集まったときや，プロジェクトの節目節目に実行します．

進め方は簡単です．参加者が持ち回りでテーマを決めながら，そのテーマについて全員が自分の意見を述べるのです．進め方のルールとしては，ただ相手の意見を受けとめることです．放っておくと議論が始まるので，参加者が安心して発言できるように，場を守る役割の人を設けます．

過去に，組み込み業界のコミュニティ「組込みシステム技術に関するサマーワークショップ(SWEST)」において実践したところ，まさに多様な意見が飛び交いました[3]．

図A　自己紹介ワークショップ
持ち回りでテーマを決め，そのテーマについて一人一人が思いや考えを述べる．ほかのメンバの発言に対して意見しないこと．議論せず，ただ受け止める．

話になってしまうのですが，きっかけは，筆者が予算申請でプロジェクタを計画に含めるのを忘れたことから始まりました．

プロジェクタの代わりに1台のディスプレイを二人で見ながら，ドキュメントをレビューします．このときの相手との距離が，プロジェクタを使うよりもディスプレイの方が近くなるわけですが，この距離の近さが，お互いに「つっこみ」を入れながら推敲していくスタイルに合っていることに気付きました．その結果，現在もプロジェクタを買わずじまいです．

さて，ペア・プログラミングで必ず話題になる効果ですが，筆者は，一人で考えるときの思考の限界を超えるものとして，二人(ペア開発)の方がより価値の高い成果を生み出していると確信しています．特に，アイデア出しや調査結果の分析，不具合の発見などで活用しています．ただし，いつもペアで作業するのではなく，時には一人ずつのアイデアを持ち寄ってから始めるなど，メンバ間で持たれ合わないように注意しています(右掲のコラム「考えるスタイルを使い分ける」を参照)．

○月×日 (火)

- 事実：昨日，悩んだところは，思い込みだった．
- 発見：思い込んでしまうとそれしか考えられなくなる．
- 教訓：頭の中だけで考ず，状況を書き出して整理する．
- 宣言：頭の中で考えていることを具体的に書き出してみて考える．

図8　4行日記の例

● 4行日記

筆者のチームの成長に大きな効果を上げているのが，この「4行日記」です．「4行日記」とは，小林惠智氏の著作で紹介されているもので，以下の四つの枠組みを利用した日記です[2]．

- 事実
- 発見(気付き)
- 教訓(学び)
- 宣言

筆者のチームでは毎日この4行日記を書いて，チーム・メンバで共有しています(**図8**)．前述した通り，私たちの開発はより高度で複雑な課題に囲まれています．そのような開発に向き合っているので，毎日，何かしらの新しい発見があるでしょう．もしくは失敗もあることでしょう．それらをノウハウとしてチーム・メンバの間に流通させ，明日の開発に，ほかのメンバに生かしていければ，チーム全体としてより高度な課題に取り組むことができます．また，一つの出来事を複数のメンバの視点でとらえることにより，

COLUMN
考えるスタイルを使い分ける

打ち合わせをしていて，意見が出ないときや，なかなかまとまらないときは，考えるスタイルを意識して使い分けましょう．ここでは二つのスタイルを紹介します．

●発散と収束

会議で意見がまとまらないことがあると思います．そのようなときは，意見を出す時間（発散）と，意見をまとめる時間（収束）に区切り，頭のスイッチを切り替えて進めましょう（**図B**）．発散と収束，それぞれの活動に集中することで，効率的に進めることができます．このとき，進行を守る役割を設けて，メンバが内容に集中できるとスムーズに進めることができます．

図B　発散と収束
時間を区切り，まずは意見を出し尽くすのに集中する（発散）．その後，意見を分析・整理してまとめることに集中する（収束）．

●考えるのに適した人数

チームで意見を出し合うときでも，まずはメンバ一人一人が考え，独自の意見を持つことから始めます．そして，その意見をより理解しやすくしたいときには，ペアを組み，相手にその意見を説明して，相手の理解を確認しながら推敲します．また，意見をよりふくらませたいときには3人で議論します．

昔から「三人寄らば文殊の知恵」とことわざで言われるように，また経験的に，3人は知恵を出し合うのに適した人数のようです．これが2人だと行き詰まることが多かったり，逆に4人以上になると議論に参加しない人が現れたりします．また，より多様な意見を出したいときには，3人のチームを複数作って議論すると効果的です．

考え方の偏りに気付く効果もあります．

この「4行日記」は開始当初，「気付き」と「学び」に重点を置いていました．しかし1年たった時点で，予想外の効果に気付きました．それは「宣言」がチームに「意志の力」をもたらしていることです．何かを実現したいというモチベーションは，個人の内側からわき上がるものです．他人が直接に働きかけられるものでもなく，自分でさえコントロールできるのか分かりません．それが，チーム・メンバの意志に触れることで，新しい意志が生まれていたのです．

＊　＊　＊

ここまで，チームの力を発揮して開発を進める考え方と，筆者のチームにおける実践例を紹介してきました．これを読み終えた皆さんに，ぜひ実践していただきたいことがあります．まず，皆さんのチームで，今，必要としていることを感じることです．

皆さんの開発チームでうまくいっていることは何でしょうか．当然のごとく行われていることで，おかしいと感じていたこと，心配になっていたことはありませんか．少し時間をとり，くまなく洗い出してみてください．そして，そのことについて思うことや考えることを，チーム・メンバとじっくり話し合ってください．チームの良い点をさらに伸ばすために，チームの改善点を克服するために，本章を活用していただければ幸いです．

参考・引用*文献

(1) 篠田義明；ビジネス文完全マスター術，角川書店，2003年9月．
(2)* 小林惠智；一日5分奇跡を起こす4行日記，オーエス出版，2002年11月．
(3) 組込みシステム技術に関するサマーワークショップ（SWEST）；SWEST7 分科会「SWEST しゃべり場」議事録，http：//www.ertl.jp/SWEST/SWEST7/report/S1-4-2.html

あなだ・けいじゅ
キャッツ（株）

<筆者プロフィール>
穴田啓樹．PHP 認定ビジネス・コーチ．メーカで組み込み開発に携わる傍ら，設計文書の改善に力を入れる．その後，プロジェクトを任されるようになるとコミュニケーションの影響の大きさに気付き，コーチングやファシリテーションを普段の開発で実践する．2005年にキャッツに入社し，新組織を設立．車載ソフトウェアの検証に向き合っている．